Statistical Methods for Data Science

Mini Project #3

Yagna Srinivasa Harsha Annadata

Yxa210024

Problem 1:

(A)

To compute the MSE (mean squared error) of an estimator (θ’) using Monte-Carlo simulation, we need to estimate for each sample by simulating random samples from the population. Then we compute the squared difference between estimate (θ’) and the true value (θ), take the average across all the samples and we obtain the MSE.

(B)

```
#code to calculate MOM and MLE
> Calculate_MLE_MOM <- function(n, theta) {
+ Sample = runif(n, min=0, max=theta)
+ MOM_Esti = 2*mean(Sample)
+ MLE_Esti = max(Sample)
+ return(c(MLE_Esti,MOM_Esti))
+ }
>
> #code to calculate the MSE
> MSE_Esti = function(n, theta) {
+ estimate = replicate(1000, Calculate_MLE_MOM(n, theta))
+ estimate = (estimate - theta)^2
+ estimate.MOM_Esti = estimate[c(TRUE, FALSE)]
+ estimate.MLE_Esti = estimate[c(FALSE, TRUE)]
+ return(c(mean(estimate.MLE_Esti),mean(estimate.MOM_Esti)))
+ }
>
> MSE_Esti(1,1)
[1] 0.3201728 0.3298294
> MSE_Esti(1,5)
[1] 8.423618 8.384768
> MSE_Esti(1,50)
[1] 803.9149 833.4034
> MSE_Esti(1,100)
[1] 3357.645 3238.024
> MSE_Esti(2,1)
[1] 0.1778052 0.1753118
> MSE_Esti(2,5)
[1] 4.172489 4.345351
> MSE_Esti(2,50)
[1] 398.7583 397.9505
> MSE_Esti(2,100)
[1] 1576.274 1647.233
> MSE_Esti(3,1)
[1] 0.1133796 0.1001241
> MSE_Esti(3,5)
```

```
[1] 2.708151 2.516601
> MSE_Esti(3,50)
[1] 278.6064 249.2343
> MSE_Esti(3,100)
[1] 1147.980 1040.178
> MSE_Esti(5,1)
[1] 0.06333097 0.04568486
> MSE_Esti(5,5)
[1] 1.641226 1.133013
> MSE_Esti(5,50)
[1] 152.6647 117.7215
> MSE_Esti(5,100)
[1] 689.3467 464.3145
> MSE_Esti(10,1)
[1] 0.03351973 0.01557688
> MSE_Esti(10,5)
[1] 0.9599630 0.4061318
> MSE_Esti(10,50)
[1] 78.56789 36.77286
> MSE_Esti(10,100)
[1] 322.3599 146.1992
> MSE_Esti(30,1)
[1] 0.010927362 0.001799196
> MSE_Esti(30,5)
[1] 0.29004884 0.04903987
> MSE_Esti(30,50)
[1] 27.104888 5.291204
> MSE_Esti(30,100)
[1] 111.88970 19.83637
```

(C)

Repeating (B)

```
> #For n = 1
> plot(c(1, 5, 50, 100),c(MSE_Esti(1,1)[1], MSE_Esti(1,5)[1],
MSE_Esti(1,50)[1], MSE_Esti(1,100)[
1]), type="b", col="blue", main="For n = 1", xlab="Theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(1,1)[2], MSE_Esti(1,5)[2],
MSE_Esti(1,50)[2], MSE_Esti(1,100
)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

```
> #For n = 2
> plot(c(1, 5, 50, 100),c(MSE_Esti(2,1)[1], MSE_Esti(2,5)[1],
MSE_Esti(2,50)[1], MSE_Esti(2,100)[
1]), type="b", col="blue", main="For n = 2", xlab="theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(2,1)[2], MSE_Esti(2,5)[2],
MSE_Esti(2,50)[2], MSE_Esti(2,100
)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

```
> #For n = 3
> plot(c(1, 5, 50, 100),c(MSE_Esti(3,1)[1], MSE_Esti(3,5)[1],
MSE_Esti(3,50)[1], MSE_Esti(3,100)[
1]), type="b", col="blue", main="For n = 3", xlab="theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(3,1)[2], MSE_Esti(3,5)[2],
MSE_Esti(3,50)[2], MSE_Esti(3,100
)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

```
> #For n = 5
> plot(c(1, 5, 50, 100),c(MSE_Esti(5,1)[1], MSE_Esti(5,5)[1],
MSE_Esti(5,50)[1], MSE_Esti(5,100)[
1]), type="b", col="blue", main="For n = 5", xlab="theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(5,1)[2], MSE_Esti(5,5)[2],
MSE_Esti(5,50)[2], MSE_Esti(5,100
)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

```
> #For n = 10
> plot(c(1, 5, 50, 100),c(MSE_Esti(10,1)[1], MSE_Esti(10,5)[1],
MSE_Esti(10,50)[1], MSE_Esti(10,1
00)[1]), type="b", col="blue", main="For n = 10", xlab="theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(10,1)[2], MSE_Esti(10,5)[2],
MSE_Esti(10,50)[2], MSE_Esti(10
,100)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

```
> #For n = 30
> plot(c(1, 5, 50, 100),c(MSE_Esti(30,1)[1], MSE_Esti(30,5)[1],
MSE_Esti(30,50)[1], MSE_Esti(30,1
00)[1]), type="b", col="blue", main="For n = 30", xlab="theta", ylab="MSE")
> lines(c(1, 5, 50, 100), c(MSE_Esti(30,1)[2], MSE_Esti(30,5)[2],
MSE_Esti(30,50)[2], MSE_Esti(30
,100)[2]), type="b", col="red")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("blue",
"red"))
```

#Graphs for varying n and fixed theta

```
> #For theta = 1
> plot(c(1, 2, 3, 5, 10, 30),c(MSE_Esti(1,1)[1], MSE_Esti(2,1)[1],
MSE_Esti(3,1)[1], MSE_Esti(5,1
)[1], MSE_Esti(10,1)[1], MSE_Esti(30,1)[1]), type="b", col="red", main="For
theta = 1", xlab="the
ta", ylab="MSE")
> lines(c(1, 2, 3, 5, 10, 30), c(MSE_Esti(1,1)[2], MSE_Esti(2,1)[2],
MSE_Esti(3,1)[2], MSE_Esti(5
,1)[2], MSE_Esti(10,1)[2], MSE_Esti(30,1)[2]), type="b", col="blue")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red",
"blue"))
```

```
> #For theta = 5
> plot(c(1, 2, 3, 5, 10, 30),c(MSE_Esti(1,5)[1], MSE_Esti(2,5)[1],
MSE_Esti(3,5)[1], MSE_Esti(5,5
)[1], MSE_Esti(10,5)[1], MSE_Esti(30,5)[1]), type="b", col="red", main="For
theta = 5", xlab="theta", ylab="MSE")
> lines(c(1, 2, 3, 5, 10, 30), c(MSE_Esti(1,5)[2], MSE_Esti(2,5)[2],
MSE_Esti(3,5)[2], MSE_Esti(5
,5)[2], MSE_Esti(10,5)[2], MSE_Esti(30,5)[2]), type="b", col="blue")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red",
"blue"))
```

```
> #For theta = 50
> plot(c(1, 2, 3, 5, 10, 30),c(MSE_Esti(1,50)[1], MSE_Esti(2,50)[1],
MSE_Esti(3,50)[1], MSE_Esti(
5,50)[1], MSE_Esti(10,50)[1], MSE_Esti(30,50)[1]), type="b", col="red",
main="For theta = 50", xl
ab="theta", ylab="MSE")
> lines(c(1, 2, 3, 5, 10, 30), c(MSE_Esti(1,50)[2], MSE_Esti(2,50)[2],
MSE_Esti(3,50)[2], MSE_Est
i(5,50)[2], MSE_Esti(10,50)[2], MSE_Esti(30,50)[2]), type="b", col="blue")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red",
"blue"))
```

```
> #For theta = 100
> plot(c(1, 2, 3, 5, 10, 30),c(MSE_Esti(1,100)[1], MSE_Esti(2,100)[1],
MSE_Esti(3,100)[1], MSE_Es
ti(5,100)[1], MSE_Esti(10,100)[1], MSE_Esti(30,100)[1]), type="b", col="red",
main="For theta = 1
00", xlab="theta", ylab="MSE")
> lines(c(1, 2, 3, 5, 10, 30), c(MSE_Esti(1,100)[2], MSE_Esti(2,100)[2],
MSE_Esti(3,100)[2], MSE_
Esti(5,100)[2], MSE_Esti(10,100)[2], MSE_Esti(30,100)[2]), type="b",
col="blue")
> legend("bottomright", legend = c("MLE", "MOM"), text.col = c("red",
"blue"))
```

(D)

Based on the plot, we can see that for all values of n and θ, the maximum likelihood estimator (θ^1 = X(n)) has a smaller MSE than the method of moments estimator (θ^2 = 2X). This indicates that the maximum likelihood estimator is generally better than the method of moments estimator for estimating θ in a Uniform(0, θ) population. The difference in MSE between the two estimators depends on both n and θ. If n increase, the difference in MSE between the two estimators becomes smaller and the maximum likelihood estimator is less dominant. If θ increases, the difference between the two estimators becomes larger, maximum likelihood estimator is more dominant.

Problem 2:

(A)

(B)

(C)

```
#Function that returns negative log-likelihood value
> logLike <- function(par, x) {
+ logLike = length(x)*log(par)-(par+1)*sum(log(x))
+ return(-logLike)
+ }
>
> #Optim function to minimize the negative log-likelihood value
> optim(par=1, fn=logLike,method = "L-BFGS-B", hessian=TRUE, lower=0.01,
x=c(21.42,14.65,50.42,28
.78,11.23))
$par
[1] 0.3236796
```

The obtained value matches with the value obtained in (B).

(D)

```
#Standard Error
> x<- optim(par=1, fn=logLikelihoodfn,method = "L-BFGS-B", hessian=TRUE,
lower=0.01, d=c(21.42,14
.65,50.42,28.78,11.23))
> standardError <- (1/x$hessian)^(1/2)
> #Confidence interval
> x$par + c(-1,1)*standardError*qnorm(0.975)
[1] 0.03996984 0.60738939
```

The approximations are only valid under the assumption that MLE is asymptotically normal, which may not be the case for small sample sizes. In this case, we only have n=5, so the approximations is not accurate.